



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/651,328	08/28/2003	Amy H. Kang	5681-69401	7788
58467	7590	04/23/2009	EXAMINER	
MHKKG/SUN P.O. BOX 398 AUSTIN, TX 78767			WANG, RONGFA PHILIP	
			ART UNIT	PAPER NUMBER
			2191	
			MAIL DATE	
			04/23/2009	PAPER
			DELIVERY MODE	

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>	
	10/651,328	KANG ET AL.	
	<b>Examiner</b>	<b>Art Unit</b>	
	PHILIP WANG	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) Responsive to communication(s) filed on 12 February 2009.
- 2a) This action is **FINAL**.                    2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) Claim(s) 1,4-12,31-39,42-44,46-51,54-56 and 58-62 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) Claim(s) \_\_\_\_\_ is/are allowed.
- 6) Claim(s) 1,4-12,31-39,42-44,46-51,54-56, and 58-62 is/are rejected.
- 7) Claim(s) \_\_\_\_\_ is/are objected to.
- 8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on \_\_\_\_\_ is/are: a) accepted or b) objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) All    b) Some \* c) None of:  
 1. Certified copies of the priority documents have been received.  
 2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)                     | 4) <input type="checkbox"/> Interview Summary (PTO-413)           |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ .                                    |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)          | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____.   | 6) <input type="checkbox"/> Other: _____ .                        |

### **DETAILED ACTION**

1. This office action is in response to communication filed on 2/12/2009.
2. Claims 1, 4-12, 31-39, 42-44, 46-51, 54-56, and 58-62 remain pending.

#### ***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

1. Claims 1, 4-12, 31-39, 42-44, 46-51, 54-56, and 58-62 are rejected under 35 U.S.C. 103(a) as being unpatentable over “The Error Handling Interface (H5E)” (herein H5E-A) in view of Srivastava et al. (USPGN 2005/0160431) (herein Srivastava).

As per claim 1,

H5E-A discloses

- A processor; and a memory comprising program instructions, wherein the program instructions are executable by the processor to implement (H5E-A, p.

1, Introduction, "...within the HDF5 library...application-called API function..."; "Example: An Error Message", see error message "HDF5-DIAG: Error detected in thread 0, this shows support for the single threaded program.):

- in each thread of a threaded program, for each error generated by one or more functions executed in the thread, store an error trace element in a memory storage area in accordance with an application programming interface (API) to the error trace mechanism (p. 1, below example, "The error stack can also be manipulated by these functions..." Since there is only one thread in this case, errors recorded are specific to the thread); and
- obtain an error trace for the thread of the threaded program in accordance with the API to the error trace mechanism (H5E-A, p. 1, 2. Error Handling Operations, 2<sup>nd</sup> para., "The error stack can also be printed..."; p. 3, see `herr_t H5Ewalk()`);
- wherein an error trace includes one or more error trace elements specific to the corresponding thread, wherein each error trace element includes information describing a particular error generated during execution of the corresponding thread (H5E-A, p. 1, Example: An Error Message, where it shows multiple trace elements, "#000: H5T.c line 462").

Art Unit: 2191

H5E-A does not specifically disclose

- two or more threads of a multithreaded program and storing an error trace element in a memory area private to the thread for each of the two or more threads .

However, Srivastava discloses

- two or more threads of a multithreaded program and storing an error trace element in a memory area private to the thread for each of the two or more threads ([0003], "...Each distributed node may also maintain multiple trace logs corresponding to separate threads...")

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Srivastava into the teachings of H5E-A to include the limitation disclosed by Srivastava. The modification would be obvious to one of ordinary skill in the art to want to be able to store traces of a thread in a separate trace log so as to maintain trace information in separate thread in separate log.

As per claim 4,

the rejection claim 1 is incorporated;

H5E-A discloses

- each error trace further includes a field indicating a count of the error trace elements in the error trace (p. 3, under `herr_t` H5Ewalk "The error stack..."; and under `typedef herr_t`, "...n is

sequence number...") .

As per claim 5,

the rejection claim 1 is incorporated;

H5E-A discloses

- wherein each error trace element indicates one or more of a location where the particular error of the error trace element occurred, an error type of the particular error, and what the particular error is (p. 1, Example: An Error Message, where line number is a location and major or minor types are shown.).

As per claim 6,

the rejection claim 5 is incorporated;

H5E-A discloses

- wherein the location of the particular error includes one or more of a function name, a source file name, and a line number where the particular error occurred (p. 1, Example: An Error Message, where H5T.c is the function name).

As per claim 7,

the rejection claim 1 is incorporated;

H5E-A discloses

- wherein the program is further configured to determine from each error trace element one or more of a location where the particular error of the error trace element occurred, an error type of the particular error, and what the particular error is (p. 1, Example: An Error Message).

As per claim 8,

the rejection claim 1 is incorporated;

H5E-A discloses

- wherein the error trace mechanism is a C/C++ interface library (p. 1, Example: An Error Message, see H5T.c).

As per claim 9,

H5E-A discloses

- a processor; and
- a memory comprising program instructions, wherein the program instructions are executable by the processor to implement a library and a threaded program configured to call library functions of the library in accordance with an application programming interface(API) to the library; wherein the library function is configured to, for each thread of the threaded program, add an error trace element to an error trace for each error generated on the thread by the library functions to an error trace in a memory storage area to the thread, wherein each error trace element includes information describing a particular error generated

- during execution of the library function(H5E-A, p. 1, Introduction,  
"...within the HDF5 library...application-called API  
function..."; "Example: An Error Message", see error  
message "HDF5-DIAG: Error detected in thread 0, this  
shows support for the threaded program; p. 1, below  
example, "The error stack can also be manipulated by  
these functions..." Since there is only one thread in  
this case, errors recorded are specific to the thread);
- and wherein, after completion of the library function, the threaded program is configured to obtain an error trace for a thread corresponding to the call of the library function in accordance with the API to the library (H5E-A, p. 1, Example:An Error Message, where it shows multiple trace elements, "#000: H5T.c line 462"; p. 3, see herr\_t\_H5Ewalk() is an API to the library to obtain error trace. ).

H5E-A does not specifically disclose

- two or more threads of a multithreaded program and storing an error trace element in a memory area private to the thread for each of the two or more threads .

However, Srivastava discloses

- two or more threads of a multithreaded program and storing an error trace element in a memory area private to the thread for each of the two or

more threads ([0003], "...Each distributed node may also maintain multiple trace logs corresponding to separate threads...")

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Srivastava into the teachings of H5E-A to include the limitation disclosed by Srivastava. The modification would be obvious to one of ordinary skill in the art to want to be able to store traces of a thread in a separate trace log so as to maintain trace information in separate thread in separate log.

As per claim 10,

the rejection claim 9 is incorporated;

H5E-A discloses

- the called library function is configured to call one or more other library functions in a function call stack, wherein each of the one or more other library functions is configured to, if the library function generates an error, add an error trace element to an error trace in a memory storage area to a thread corresponding to the function call stack (see 1. Introduction) .

H5E-A does not specifically disclose

- a memory area private to the thread for each of the two or more threads .

However, Srivastava discloses

- a memory area private to the thread for each of the two or more threads ([0003],  
“...Each distributed node may also maintain multiple trace logs corresponding to  
separate threads...”)

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Srivastava into the teachings of H5E-A to include the limitation disclosed by Srivastava. The modification would be obvious to one of ordinary skill in the art to want to be able to store traces of a thread in a separate trace log so as to maintain trace information in separate thread in separate log.

As per claims 11-12,

the rejection of claim 9 is incorporated and further

- claims 11-12 recite the same limitation of claims 7 and 8 respectively and are rejected for the same reason set forth in the rejection of claims 7 and 8 respectively.

As per claim 31,

H5E-A discloses

- a processor; and a memory comprising program instructions, wherein the program instructions are executable by the processor to implement a library comprising one or more library functions and an application programming

interface (API) to the library, wherein the API includes: one or more function definitions configured for access of the one or more library functions by a threaded program; and a function definition for a get error trace function configured for access by the threaded program to get error traces generated by the one or more library functions in two or more threads of the multithreaded program, wherein each error trace is stored in a memory storage area to the thread (see page 1, sections 1 and 2, for API functions for a thread program; also see page 3, for example

```
herr_t H5Ewalk() );
```

- wherein each error trace includes one or more error trace elements specific to the thread, wherein each error trace element includes information describing a particular error generated during execution of the corresponding thread (page 1, paragraph 7, "Each thread has its own error stack...multi-threading...").

H5E-A does not specifically disclose

- a multithreaded program and storing an error trace element in a memory area private to the corresponding thread for each of the two or more threads .
- threads .

However, Srivastava discloses

- two or more threads of a multithreaded program and storing an error trace element in a memory area private to the thread for each of the two or

more threads ([0003], "...Each distributed node may also maintain multiple trace logs corresponding to separate threads...")

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Srivastava into the teachings of H5E-A to include the limitation disclosed by Srivastava. The modification would be obvious to one of ordinary skill in the art to want to be able to store traces of a thread in a separate trace log so as to maintain trace information in separate thread in separate log.

As per claim 32,  
the rejection of claim 31 is incorporated and further

- Refer to rejection of claim 10.

As per claim 33,  
the rejection of claim 32 is incorporated and further

- Refer to rejection of claim 6.

As per claim 34,  
the rejection of claim 31 is incorporated and further

- Refer to rejection of claim 7.

As per claim 35,

the rejection of claim 31 is incorporated and further

- Refer to rejection of claim 8.

As per claim 36,

H5E-A discloses a system, comprising:

- means for a plurality of functions in a function call stack to generate information describing one or more errors generated by the plurality of functions (See 1. Introduction);
- means to obtain the generated information (H5E-A, p. 1, 2. Error Handling Operations, 2<sup>nd</sup> para., "The error stack can also be printed..."; p. 3, see `herr_t H5Ewalk()`);
- and means to determine from the obtained information one or more of a location where each error occurred, an error type of each error, and what the each error is (H5E-A, p. 1, Example: An Error Message, where it shows multiple trace elements, "#000: H5T.c line 462").

H5E-A does not specifically disclose

- two or more threads of a multithreaded program and storing an error trace element in a memory area private to the thread for each of the two or more threads .

- threads .

However, Srivastava discloses

- two or more threads of a multithreaded program and storing an error trace element in a memory area private to the thread for each of the two or more threads ([0003], "...Each distributed node may also maintain multiple trace logs corresponding to separate threads...")

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Srivastava into the teachings of H5E-A to include the limitation disclosed by Srivastava. The modification would be obvious to one of ordinary skill in the art to want to be able to store traces of a thread in a separate trace log so as to maintain trace information in separate thread in separate log.

As per claim 37,

the rejection claim 36 is incorporated;

H5E-A discloses

- wherein the plurality of functions are functions of a library, further comprising means to call the plurality of functions in the function call stack from a program (p. 1. Introduction) .

As per claim 38,

the rejection claim 37 is incorporated;

H5E-A discloses

- wherein the library is a C/C++ interface library (p. 1, Example: An Error Message).

As per claim 39,

H5E-A discloses

- Calling one or more functions in a thread of a threaded program; in each thread of the threaded program, for each error generated by the one or more functions called in the thread, storing an error trace element in a memory storage area private to the thread in accordance with an application programming interface (API) to an error trace mechanism; and the program obtaining an error trace for a thread of the threaded program in accordance with the API to the error trace mechanism; wherein each error trace includes one or more error trace elements specific to the corresponding thread, wherein each error trace element includes information describing a particular error generated during execution of the corresponding thread(p. 1, below example, "The error stack can also be manipulated by these functions..." Since there is only one thread in this case, errors recorded are specific to the thread; H5E-A, p. 1, 2. Error

Handling Operations, 2<sup>nd</sup> para., "The error stack can also be printed..."; p. 3, see herr\_t H5Ewalk());H5E-A, p. 1, Example: An Error Message, where it shows multiple trace elements, "#000: H5T.c line 462");

H5E-A does not specifically disclose

- a multithread program and for each of two or more thread of the threaded program, adding an error trace element for each error generated on the thread to an error trace in a memory storage area private to the thread.

However, Srivastava discloses

- a multithread program and for each of two or more thread of the threaded program, adding an error trace element for each error generated on the thread to an error trace in a memory storage area private to the thread.
- ([0003], "...Each distributed node may also maintain multiple trace logs corresponding to separate threads..." here the threaded program is a multithreaded program)

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Srivastava into the teachings of H5E-A to include the limitation disclosed by Srivastava. The modification would be obvious to one of

ordinary skill in the art to want to be able to store traces of a thread in a separate trace log so as to maintain trace information in separate thread in separate log.

As per claim 42,

the rejection claim 39 is incorporated;

H5E-A discloses

- wherein each error trace element indicates one or more of a location where the particular error of the error trace element occurred, an error type of the particular error, and what the particular error is (p. 1, Example: An Error Message).

As per claim 43,

the rejection claim 42 is incorporated;

H5E-A discloses

- wherein the location of the particular error includes one or more of a function name, a source file name, and a line number where the particular error occurred (p. 1, Example: An Error Message).

As per claim 44,

the rejection claim 39 is incorporated;

H5E-A discloses

- further comprising determining from each error trace element one or more of a location where the particular error of the error trace element occurred, an error type of the particular error, and what the particular error is (p. 1, Example: An Error Message).

As per claim 46,

the rejection claim 39 is incorporated;

H5E-A discloses

- wherein the error trace mechanism is a C/C++ interface library (p. 1, Example: An Error Message).

As per claim 47,

H5E-A discloses

- A threaded program calling library functions of a library in accordance with an application programming interface (API) to the library a thread program and for one thread of the threaded program, adding an error trace element for each error generated on the thread by the library functions to an error trace in a memory storage area private to the thread (p. 1, below example, "The error stack can also be manipulated by these functions..." Since there is only one thread in this case, errors recorded are specific to the thread);

Art Unit: 2191

- after completion of a called library function, the threaded program obtaining an error trace for a thread corresponding to the call of the library function in accordance with the API to the library (H5E-A, p. 1, 2.

Error Handling Operations, 2<sup>nd</sup> para., "The error stack can also be printed..."; p. 3, see `herr_t H5Ewalk()`;

- a threaded program and each error trace element includes information describing a particular error generated during execution of the corresponding thread (H5E-A, p. 1, Example: An Error Message, where it shows multiple trace elements, "#000: H5T.c line 462")

H5E-A does not specifically disclose

- a multithread program and for each of two or more threads of the threaded program, adding an error trace element for each error generated on the thread by the library functions to an error trace in a memory storage area private to the thread.

However, Srivastava discloses

- a multithread program and for each of two or more threads of the threaded program, adding an error trace element for each error generated on the thread to an error trace in a memory storage area private to the thread ([0003], "...Each distributed node may also maintain multiple trace logs corresponding to separate

Art Unit: 2191

threads..." here the threaded program is a multithreaded program)

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Srivastava into the teachings of H5E-A to include the limitation disclosed by Srivastava. The modification would be obvious to one of ordinary skill in the art to want to be able to store traces of a thread in a separate trace log so as to maintain trace information in separate thread in separate log.

As per claims 48-50,

the rejection of claim 47 is incorporated and further

- claims 48-50 recite the same limitation of claims 10, 7, and 8 respectively and are rejected for the same reason set forth in the rejection of claims 10, 7 and 8 respectively.

As per claim 51,

- it is the computer-accessible medium claim corresponding to method claim 39 and is rejected for the same reason set forth in connection of the rejection of claim 39 above.

As per claim 54,

the rejection claim 51 is incorporated;

H5E-A discloses

- wherein each error trace element indicates one or more of a location where the particular error of the error trace element occurred, an error type of the particular error, and what the particular error is (p. 1, Example, An Error Message) .

As per claim 55,

the rejection claim 54 is incorporated;

H5E-A discloses

- wherein the location of the particular error includes one or more of a function name, a source file name, and a line number where the particular error occurred (p. 1, Example, An Error Message) .

As per claim 56,

the rejection claim 51 is incorporated;

H5E-A discloses

- wherein the program instructions are further computer-executable to implement determining from each error trace element one or more of a location where the particular error of the error trace element occurred, an error type of the particular

error, and what the particular error is (p. 1, Example, An Error Message) .

As per claim 58,

the rejection claim 51 is incorporated;

H5E-A discloses

- the library is a C/C++ interface library (p. 1, Example, An Error Message) .

As per claim 59,

- it is the computer-accessible medium claim corresponding to method claim 47 and is rejected for the same reason set forth in connection of the rejection of claim 47 above.

As per claims 60-62,

- they are the computer-accessible medium claims corresponding to method claims 48-50 respectively and are rejected for the same reason set forth in connection of the rejection of claims 48-50 above respectively.

## **Response to Arguments**

In the remark,

**1) The Applicant argued –**

Srivastava does not teach that the “trace logs” are error traces.

**1) Examiner’s response –**

As the applicant correctly points out that, Srivastava, [0002], discloses “...use tracing and trace logs to diagnose programs or errors...” appears to teach trace logs can be error traces. For the sake of argument, H5E-A clearly discloses error traces and the combination H5E-A’s error traces (2<sup>nd</sup> para., “The error stack can also be printed...”) with Srivastava’s trace log should clearly show error traces in the combination.

**2) The Applicant argued –**

Srivastava clearly teaches that a given log file stores traces for all threads on the corresponding node.

**2) Examiner’s response --**

Srivastava, [0003], discloses

[0003] In a parallel or distributed environment, there are potentially a number of distributed network nodes, with each node running a number of distinct execution entities such as threads, tasks or processes, which may comprise of a plurality of threads. In many modern computer applications, these threads perform complex interactions with each other, even across the network to threads on other nodes. Often, each of the distributed nodes maintains a separate log file to store traces for their respective threads.  
Each distributed node may also maintain multiple trace logs corresponding to separate threads on that node.

In this paragraph, Srivastava discloses two different embodiments of storing traces:

The first embodiment is storing traces for threads with one log file on the node (“each of the distributed nodes maintains a separate log file to store traces for their respective threads.”).

The second embodiment is storing trace logs for each thread on a node in a corresponding separate log file. So, there is one corresponding separate log file for each thread on the node. (“Each distributed node may also maintain multiple trace logs corresponding to separate threads on that node.”).

The second embodiment clearly show there are multiple trace logs for corresponding separate threads. The applicant’s argument appears to be based on the first embodiment only.

### **3) The Applicant argued –**

The combination of H5E-A and Srivastava does not teach the limitation of a memory storage area private to the thread and in accordance with an application programming interface (API) to an error trace mechanism.

### **3) Examiner’s response --**

In reference to examiner’s response in item 2, there are trace logs for corresponding separate threads. Therefore, each thread has its corresponding/separate trace log. The examiner considers the corresponding trace log for a thread is a memory storage area private to the thread. H5E-A, page 1, sections 1 and 2, for API functions for a thread program; also see page 3, for example herr\_t H5Ewalk(), appears to disclose such API .

**4) The Applicant argued –**

The reason for combine H5E-A and Srivastava is not proper.

**4) Examiner's response --**

The examiner has provided a revised reason to combine.

**5) The Applicant argued –**

The recited prior art does not have enabling embodiment of the multithreaded embodiment.

**5) Examiner's response --**

H5E-A shows at least a thread program (page 1, in the example box, "error detected in thread 0"). Srivastava discloses multiple threads as discussed in response item 2. The combination appears to disclose multiple threads.

***Conclusion***

**THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Philip Wang whose telephone number is 571-272-5934. The examiner can normally be reached on Mon - Fri 8:00 - 4:00PM. Any inquiry of general nature or relating to the status of this application should be directed to the TC2100 Group receptionist: 571-272-2100.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Wei Y Zhen/

Supervisory Patent Examiner, Art Unit 2191